(11) Japanese Patent Laid-Open No. 10-229538

(43) Laid-Open Date: August 25, 1998

(21) Application No. 9-30543

(22) Application Date: February 14, 1997

(71) Applicant: Canon KABUSHIKI KAISHA

(72) Inventor: ITO et al.

(74) Agent: Patent Attorney, Giiti MARUSHIMA

(54) [Title of the Invention] DATA COMMUNICATION SYSTEM, PRINT SYSTEM, AND DATA COMMUNICATION DEVICE

(57) [Abstract]

[Object] To prevent instructions made in the operating portions of nodes, e.g., a video device and a printer, connected to each other in a network, from interfering each other.

[Solving Means] A data communication system containing a first node (VTR2 in Fig. 1) having an image display monitor and an operating portion, and a second node (printer in the same drawing) having an operating portion and excluding an image display monitor is characterized in that the first node and the second node communicate with each other by a Ds-Ciak method, and the system contains an arbitrating means for arbitrating an instruction from the operating portion of the first node and an instruction from the operating portion

of the second node.

[Claims]

[Claim 1] A data communication system containing a first node having an image display monitor and an operating portion, and a second node having an operating portion and excluding an image display monitor, in which the first node and the second node communicate with each other via DS-Link, characterized in that the system contains an arbitrating means for arbitrating an instruction from the operating portion of the first node and an instruction from the operating portion of the second node.

[Claim 2] A data communication system according to Claim 1, characterized in that the arbitrating means prioritizes an operation of the first node over an operation of the second node.

[Claim 3] A data communication system according to Claim 1, characterized in that the arbitrating means prioritizes an operation of the second node over an operation of the first node.

[Claim 4] A data communication system according to Claim 1, characterized in that the first node and the second node are connected to each other via a serial bus.

[Claim 5] A data communication system according to Claim 4, characterized in that the serial bus is in conformity to 1394 Standard.

[Claim 6] A data communication system according to Claim 1, characterized in that the second node is a printer.

[Claim 7] A data communication system according to Claim 6, characterized in that the arbitrating means disables an operation made in the operating portion of the printer during printing with the printer.

[Claim 8] A print system containing a video device having an image display monitor and an operating portion, and a printer having an operating portion, characterized in that the system contains an arbitrating means for arbitrating an instruction from the operating portion of the video device and an instruction from the operating portion of the printer.

[Claim 9] A print system according to Claim 8, characterized in that the video device transmits video data by Isochronous transfer, and receives information on the state of the printer by Asynchronous transfer.

[Claim 10] A print system according to Claim 8, characterized in that the information representing the state of the printer is displayed on the monitor means.

[Claim 11] A data communication device constituting the first node or the second node of the data communication system.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention] The present invention relates to a data communication system and a print system data communication device, and in particular to a system in which data communication is carried out between respective devices.

[0002]

[Description of the Related Art] Hard disks and printers are most fluently used in the peripherals of personal computers. For data communication, these peripherals are connected to personal computers via SCSI or the like which is a multipurpose interface for small-scale computer, and also, is a typical digital interface.

[0003] Moreover, digital cameras and digital video cameras are means for inputting to personal computers (hereinafter, abbreviated as PC) which are ones of peripherals. In recent years, techniques have been advanced in the field where images such as still images and moving images picked-up by digital cameras and video cameras are captured into PC, and then, are stored in hard disks or edited and, thereafter, color-printed by means of printers. The number of users in the field has increased.

[0004] When the captured image data are output from the PC to printers or hard disks, data communication is made via the above-mentioned SCSI or the like. In such cases, to transmit information having a large data amount such image data, it is necessary to employ interfaces having a high transfer data rate and being effective for multi-purposes, as the above-mentioned digital I/F.

[0005] As described above, conventionally, the respective devices are connected to PC as a host. Image data picked up by a camera is printed by way of the PC.

[0006] However, in some of the above-described SCSIs, the transfer data rate is low, and the used cables are thick. Moreover, they are restricted to the types and number of connected peripherals, the connection systems, and so forth. Thus, it is pointed out that they are disadvantageous in various respects.

[0007] Most of the general home PCs have connectors provided on the back sides thereof, to which SCSIs and other cables are connected. Moreover, the shape and type of such connectors are diversified. Thus, the plugging on or off is not easy. For connection of moving or portable types of digital cameras and digital video cameras, which are stationary devices in normal cases, it is necessary to connect them to the backside connectors of PCs. This is troublesome.

[0008] In ordinary cases, many peripherals are connected to a personal computer. In future, the number of types of peripherals will increase. Moreover, communication between many digital devices in addition to the PC peripherals, in connection to networks, will be possible. This will be very convenient. However, the communication of data having a large amount between devices will be frequently made, which causes the networks to congest. This may affect the communication between the other devices within the networks. For example, in the case of data communication between PC and a printer, e.g., when a user wants to print continuously or quickly, communication between devices not recognized by the user will exert an influence over the whole network or the PC or the like as a host, so that the image can not correctly be printed or the print is delayed. Thus, troubles will occur in print data communication or the like, depending on the charge to the PC caused by the congestion of the network and the operation condition of the PC.

[0009] To solve the above-descried problems, according to the present invention, there is realized data communication between devices such as PC, printers and the other peripherals, digital cameras, camera-integrated digital VTR or the like which are connected in a network configuration, using a multipurpose digital I/F (e.g., IEEE1394-1995 high performance serial bus) which can be systematically mounted in the respective devices and solve most of the problems of conventional digital I/Fs, and also there is realized so-called direct print in which image data is transferred from a digital camera or a camera-integrated digital VTR to a printer, and is printed.

[0010] However, in the above-described case, operations made in the operating portions on the printer side and on the camera side cannot be arbitrated.

[0011]

[Means for Solving the Problems] The data communication system of the present invention containing a first node having an image display monitor and an operating portion, and a second node having an operating portion and excluding an image display monitor, in which the first node and the second node communicate with each other via DS-Link is characterized in that the system contains an arbitrating means for arbitrating an instruction from the operating portion of the first node and an instruction from the operating portion of the second node.

[0012]

[Embodiments]

<First Embodiment> Hereinafter, a first embodiment will be described with reference to the drawings.

[0013] Fig. 2 shows an example of the configuration of a network which embodies the present invention.

[0014] In this embodiment, an IEEE1394 serial bus is used as a digital I/F. Thus, first, the IEEE1394 serial bus will be described.

[0015] <<Outline of Techniques of IEEE1394 Serial Bus>> With the advent of home VTRs and DVDs, it has been necessary to support the transfer of a large information-amount of data such as video-date, audio-data, or the like in real time. To transfer such video-data, audio-data, or the like in real time, and cause PCs to read the data, and transfer the date to other devices, interfaces capable of transferring data at a high speed and provided with necessary transfer functions are required. One of the interfaces develop from the above-described points of view, is the IEEE1394-1995 (High Performance Serial Bus) (hereinafter, abbreviated as 1394 serial bus).

[0016] Fig. 7 is an example of a network system configured using the 1394 serial bus. This system is provided with devices A, B, C, D, E. F, G, and H. The devices A and B, A and C, B and D, D and E, C and F, C and G, and C and H are connected with each other, respectively, via the twist pair cables of the 1394 serial buses. For examples, the devices are a digital VTR, DVD, a digital Camera, a hard disk, a monitor, and the like.

[0017] A daisy chain mode and node branch mode may be mixed for use as connection modes between the devices. Thus, the connection having a high degree of freedom can be realized.

[0018] The respective devices have IDs inherent herein. The respective devices recognize each other, and thereby, one network is formed as far as the devices are connected with each other via the 1394 serial buses. The respective devices play a role as relays, and wholly, configure one network only by sequentially connecting the devices between them via one 1394 serial bus, respectively. Moreover, due to the plug & play function, which is one of characteristics of the 1394 serial bus, the respective devices have a function to automatically recognize the devices and the connection conditions at the time when the devices are connected via the cables.

[0019] In the system shown in Fig. 7, when a device is removed from the network, or a new device is added, the bus resetting is automatically carried out. Thus, the network configuration before the resetting is reset, and then, a new network is re-constructed. Due to this function, the configuration of the network can be set and recognized at any time.

[0020] The data transfer rates of 100/200/400 Mbps are provided. A device having a higher transfer rate supports a device having a lower transfer rate to become compatible with it.

[0021] Referring to the data transfer mode, an asynchronous transfer mode for transferring not-synchronous data (Asynchronous data, hereinafter, abbreviated as Async data) such as a control signal or the like, and an isochronous transfer mode for transferring synchronous data (Isochronous data, hereinafter, abbreviated as Iso data) such as video data, audio data, or the like in real time are available.

In each cycle (ordinarily, one cycle is 125 μs), a cycle start packet (CSP) is transferred, and subsequently, the Async data and the Iso data are concurrently transferred while the transfer of the Iso data is caused to take precedence.

[0022] Then, Fig. 8 shows the components of the 1394 serial bus.

[0023] As a whole, the 1394 serial bus is formed so as to have a layer structure (hierarchy). As shown in Fig. 8, the cable of the 1394 serial bus shows a characteristic of hardware at a heist degree. A connector port with which the connector of the cable is connected is provided thereon. A physical layer and a link layer as hardware are provided thereon.

[0024] The hardware portion is composed substantially of an interface chip. The physical layer carries out coding, control relevant with connectors, and so forth. The link layer carries out the transfer of a packet, the control of cycle time, and so forth.

[0025] The transaction layer of the firmware carries out the control of data to be transferred (transacted), and outputs an instruction such as Read or Write. The serial bus management portion manages the connection state of the respective connected devices and ID, and also manages the configuration of the network.

[0026] The hardware and the firmware substantially constitute the 1394 serial bus.

[0027] The application layer of the software portion becomes different depending on used software, and specifies what kind of data should be applied on the interface, and is specified by a protocol such as an AV protocol.

[0028] The configuration of the 1394 serial bus is configured as described above.

[0029] Hereinafter, Fig. 9 shows the address space of the 1394 serial bus.

[0030] The respective devices (nodes) connected via the 1394 serial buses are caused to have 64 bit addresses inherent in the nodes without fail. The addresses are stored in ROM. Thereby, the node addresses of one party and the other party can recognized at any time. Thus, communication of which the destination is specified can be made.

[0031] The addressing via the 1394 serial buses is made in conformity to the IEE1212 Standard. As for the address setting, the first 10 bits are used for selection of the number of a bus, the next 6 bits for selection of a node ID number, and the remaining 48 bits is an address width given for the respective devices. Thus, they can be used as the respective inherent address spaces. The final 28 bits are used as an inherent data area to store information on the identification and use-conditions of the respective devices, or the like.

[0032] The outline of the techniques of the 1394 serial bus is described above.

[0033] Hereinafter, the part of the techniques characteristic of the 1394 serial bus will be described in more detail.

[0034] <<Electrical Specifications of 1394 Serial Bus>> Fig. 10 shows the cross-section of the 1394 serial bus.

[0035] As for the 1394 serial bus, an electric source line is provided in the connection cable, in addition to two sets of twisted pair signal lines. Thereby, electric power can be supplied to a device having no electric source or a device of which the power is reduced due to a fault.

[0036] The voltage of an electric source supplied through the electric source line is specified to be in the range of 8 to 40V, while the current is specified to be at a maximum current DC 1.54 A.

[0037] <<DS-Link Encoding>> Fig. 11 illustrates the DS-Link encoding system of a data transfer format which is adopted in the 1394 serial bus.

[0038] The DS-Link (Data/Strobe Link) encoding system is adopted in the 1394 serial bus. This DS-Link encoding system is suitable for high speed serial communication. The configuration requires two signal lines. One of the twisted paired lines is mainly used to transmit data, and the other twisted paired line is used to transmit a strobe signal.

[0039] On the reception side, the data to be caused to communicate and the strobe are exclusive-ORed to re-produce to re-create a clock.

[0040] Merits of the use of the DS-Link encoding system lie in that the transfer efficiency is high ac compared with the other serial data transfer systems. The circuit-scale of a controller LSI can be reduced, since a PLL circuit becomes unnecessary. Moreover, since it is unnecessary to send information for showing the idle-state when no date to be transferred exists, the transceiver circuit of the respective devices can be put into the sleep state, so that the power consumption can be reduced, and so forth.

[0041] <<Sequence of Bus Resetting>> Referring to the 1394 serial bus, the respective connected devices (nodes) are given IDs, and thus, the network configuration is recognized.

[0042] When the network configuration is changed, e.g., when an node is extracted off, the number of nodes is increased or decreased due to the ON/OFF of the electric source, or the like, and thus, it is required to recognize a new network configuration, the respective nodes detect such changes, transmit a bus-reset signal to the bus, and enter in the mode by which the new network is recognized. The method of detecting such changes is performed by the detection of a bias voltage on the 1394 port base.

[0043] A bus rest signal is transmitted from a node. The physical layer of each node receives the bus reset signal, simultaneously, inform the link layer on the generation of the bus reset signal, and transmits the bus reset signal to the other nodes. Finally, after all of the nodes receive the bus reset signal, the bus reset starts.

[0044] The bus rest can be started by the above-described extraction of a cable, for the detection of the hardware made due to an abnormality in the network or the like, and by an instruction made directly to the physical layers, which is carried out by the host control according to a protocol.

[0045] Moreover, when the bus rest starts, the data transfer is temporarily interrupted, so that the data transfer is kept waiting during the interruption. After the bus resetting, the data transfer starts again in the new network configuration.

[0046] The sequence of the bus resetting is described above.

[0047] <<Sequence of Determination of Node ID>>

After the bus is reset, the process starts, in which ID is given to the respective nodes so that the nodes can configure a new network. The general sequence from the bus-rest to the determination of node IDs will be described with reference to the flowcharts of Figs. 19, 20, and 21.

[0048] The flowchart of Fig. 19 illustrates a series of bus works in the course of from the generation of bus-reset, to the determination of node IDs, and to enabling of data transfer.

[0049] First, at step S101, it is monitored at any given point in time whether bus-rest generates in the network is not. If the bus-reset occurs due to the ON/OFF of an electric source or the like, the process is moved to step S102.

[0050] At the step S102, parentages between the respective nodes directly linked to each other are declared to know the connection state of a new network which is formed from the reset network. If the parentages are determined between all of the nodes at step S103, one route is determined at step S104. The declaration of the parentages at the step 102 is carried out until the parentages are determined between all of the nodes, and also, an route is not determined till then.

[0051] If the route is determined at the step S104, the node ID setting work for giving IDs to the respective nodes is carried out at step 105. The node IDs are set in a predetermine node-sequence. The setting work is repeated until the IDs are given to all of the nodes. Finally, the setting of the IDs to all of the nodes is completed at step S106. Then, the new network configuration is recognized by all of the nodes. Thus, at step S107, data-transfer between the nodes becomes possible, and the data transfer is started.

[0052] When the process gets into the state of the step S107, the mode in which the generation of bus-reset is monitored is started. If the bus-reset occurs, the setting works of from the step S101 to the step S106 is repeated again.

[0053] The flowchart of Fig. 19 is described above. Figs. 20 and 21 are the flowcharts showing in more detail the processes of from the bus-resetting to the determination of the route, and the processes till the completion of the ID-setting after the determination of the route, respectively.

[0054] First, the flowchart of Fig. 20 will be described.

[0055] If bus-rest occurs at step S201, the network configuration is reset. It is to be noted that the occurrence of the bus-reset is monitored at any given point in time at the step S201.

[0056] Then, at step S202, as a first step of the work in which the connection state of the reset network is recognized again, flags for representing leaves (nodes) are set on the respective devices. Moreover, at step S203, it is checked how many ports of each device are connected to other nodes.

[0057] To start the declaration of a parentage, corresponding to the result of the number of ports obtained at step 204, the number of undefined (the parentage has not been determined yet) ports is checked. The number of ports = the number of undefined ports is effective directly after the bus reset. As more parentages are determined, the number of undefined ports detected at the step S204 changes.

[0058] First, parentages can be declared for leaves only, immediately after the bus-reset. It can be known by checking the number of ports at the step S203 whether the nodes are leaves or not. At step S205, a leaf declares it to the other node connected to the leaf that "I am a child, and you are a parent", and thus, the process is finished.

[0059] As for a node for which it is recognized that it has plural ports and is a branch at the step S203, the number of undefined ports > 1 is effective immediately after the bus-rest, so that the process is moved to step S206, and a flag for representing "a branch" is set on the node. At step S207, the node awaits in order to accept "a parent" in the declaration of the parentage made by the leaf.

[0060] For the branch which, at step S207, accepts the declaration of a parentage made by the leaf, the number of undefined ports is confirmed, if necessary. If the number of undefined ports is 1, the branch can made the declaration "I am a child" for the node connected to the remaining port, at the step S205. As for a branch having the number of undefined ports of 2 or more, which is confirmed at the step S205, at the second time or later, the branch awaits in order to accept "a parent" made by the leaf or another branch at the step S207.

[0061] Finally, if one branch or exceptionally a leaf (since the leaf does not quickly operate, although the leaf can declare that "I am a child") shows the number of undefined ports of zero as a confirmation result at the step S204, this means that the declaration of parentages in the whole network is completed. For only one node having the number of undefined ports of zero (the number is determined for the ports of a parent in any case), a route flag is set as step S208. Thus, the node is recognized as the route at step S209.

[0062] Thus, the processes ranging from the bus-reset to the declaration of parentages between all of the nodes in the network, as shown in Fig. 20, are completed.

[0063] Hereinafter, the flowchart of Fig. 21 will be described.

[0064] First, for the respective nodes, flag information are set as the leaves, the branches, and the routes, in the sequence of Figs. up to 20. They are classified based on the flag information at step S301.

[0065] Referring to the work for rendering IDs to the respective nodes, a leaf is the first node for which ID can be set. IDs are set for the nodes having numbers (node numbers, from 0), in the order toward a larger number in the sequence of leaves, branches, and then, routes.

[0066] At step S302, the number N (natural number) of the leaves existing in the network is set. Thereafter, at step S303, each leaf requests the route to give ID to it. In the case where plural requests are made, the route makes the arbitration at step S304 (work for arbitrating the requests so that one node remains). At step S306, an ID number is rendered to one node which win the arbitration. The fail result is posted to a node which loses the arbitration. The leaf which fails to acquire ID at the step S306 requests for ID again. The above-described processes are repeated. At step S307, from the leaf acquiring ID, the ID information of the node is transferred to all of the nodes by broadcast. The broadcast of the ID information of the one node is completed. Thus, at step S308, the number of the remaining leaves is reduced by one. At step S309, if the number of the remaining leaves is 1 or more, the work of requesting ID at the step S303 and the succeeding processes are repeated. Finally, if all of the leaves broadcast the ID information, N = 0 is effective at the step S309. Thereafter, the ID setting of the branches is started.

[0067] The ID setting of the branches is made in the same manner as that of the leaves.

[0068] At step S310, first, the number M (natural number) of the branches existing in the network is set. Thereafter, at step S311, each branch requests a route to give ID to it. For the requests, the route makes the arbitration at step S312. At step S306, the larger number next to the number rendered to the leaf at last is given to the branch which wins the arbitration first. Thus, the numbers are sequentially given to the branches which win the arbitration in the order toward a larger number. At step S313, the route posts to the branches which request ID, the ID information or the fail result. The leaf which fails to acquire ID at the step S314 requests for ID again. The above-described processes are repeated. At step S315, from the branch which acquires ID, the ID information of the node is transferred to all of the nodes by broadcast. The broadcast of The ID information of the one node is completed. Thus, at step the S316, the number of the remaining branches is reduced by one. At step S317, if the number of the remaining branches is 1 or more, the work of requesting ID at the step S303 and the succeeding processes are repeated until all of the branches broadcast the ID information. When all of the branches acquire node IDs, M = 0 is effective at the step S317. Thus, the branch ID acquiring mode is completed.

[0069] At the completion, the node which finally does not acquires ID information is the routes only. Thus, at step S318, for the route, the smallest one of the numbers not given is set as the route ID number, and broadcasts the route ID information at step S319.

[0070] Thus, as seen in Fig. 21, the procedures ranging from the determination of parentages and the setting of the IDs for all of the nodes are completed.

[0071] Hereinafter, operations in a practical network shown as an example in Fig. 12 will be described with reference to Fig. 12.

[0072] Fig. 12 is explained. In the hierarchical structure, a lower-level node A and a lower-level node C are connected directly to a node B, a lower-level node D is connected directly to a node C, and moreover, a lower level node E and a lower level node F are connected directly to a node D. Hereinafter, the hierarchical structure, the route node, and the procedure for determining node ID will be described below.

[0073] To recognize the connection-state of the nodes after the bus is reset, parentages are declared between the directly-connected ports of the respective nodes. It may be defined that, in a parentage, a node at an higher level is called a parent, and a node at a lower level is called a child in a hierarchical structure.

[0074] Referring to Fig. 12, the node A declares the parentage first. Basically, a node (called a leaf) having only one connected-port can declare a parentage first. The leaf can know that it has only one connected port. Thereby, the leaf can recognize that it positions at the end of the network. Of such nodes, a parentage is declared first by the node which operates most quickly, then by the node which operates next to the said node, and so on. The port on the side where the parentage is declared as described above (node A in the case of between A and B) is set as a child, and the port on the other part (node B) is set as a parent. Thus, a parentage of child - parent is determined between the nodes A - B, a parentage of child - parent between the nodes E - D, and a parentage of child - parent between the node F - D.

[0075] At a higher level by one layer, in the nodes having plural connected ports, a node for which another node declares "a parentage" declares a parentage for a node at a still higher level. Thus, the nodes sequentially declare parentages. In Fig. 12, first, for the node D, a parentage is determined between D - E and between D - F. Thereafter, the node D declares a parentage for the node C. As a result, a parentage of child - parent is determined between the nodes D - C.

[0076] The node C which receives the declaration of parentage from the node D declares a parentage for the node B which is connected to the other port of the node C. Thus, a parentage of child - parent is determined between the nodes C - B.

[0077] The hierarchical structure shown in Fig. 12 is formed as described above. Finally, the node B which is a parent for all of the nodes which are connected to the node B is determined as a route node. Thus, a route node exists in one network structure.

[0078] In Fig. 12, the node B is determined as a route node. However, if the node B which receives the declaration of a parentage from the node A declares a parentage for other nodes earlier, a node other than the node B may be the route. That is, there is a possibility with which any node becomes a route node. Thus, the route node may be different even in the same network structure.

[0079] After the route node is determined, the process enters the mode for determining the respective node IDs. In this case, all of the nodes post their determined IDs to all of the other nodes, respectively (broadcast function).

[0080] The ID information of a node includes its own node number, information on the connected position, the number of ports contained in the node, the number of connected ports, information on the parentage of each port, and so forth.

[0081] Referring to the procedure for allocating the node ID numbers, first, a node (leaf) having only one connected port can be started. Node numbers = 0, 1, 2, , are sequentially allocated to the leaves.

[0082] A node which acquires a node ID number transmits information including the node ID number to the respective other nodes by broadcast. Thereby, the ID number is recognized as "already allocated".

[0083] After all of the leaves acquire their own ID numbers, the process moves to the branches. The node ID numbers succeeding those of the leaves are allocated to the respective nodes. Similarly to the case of the leaves, the branches to which their own ID information are allocated sequentially broadcast the node ID information. Finally, the route node broadcasts its own ID information. That is, the route possesses a maximum node ID number.

[0084] Thus, the allocation of the node IDs in the whole hierarchical structure is completed. The network is re-constructed. The initialization of the bus is finished.

[0085] << Arbitration >> In the case of the 1394 serial bus, it is necessary to carry out the arbitration of a bus use-license before data is transferred. In the logical bus network, the individual devices connected to the 1394 serial bus relay a transferred signal so as to transmit the same signal to all of the devices in the network. Accordingly, to prevent packets from interfering each other, the arbitration is necessary. Thereby, only one device can transfer data in a certain time-period,

[0086] Figs. 13(a) and 13(b) show that the use of the bus is requested and that the use of the bus is permitted, respectively, for illustration of the arbitration. Hereinafter, the arbitration is described with reference to the drawings.

[0087] When the arbitration is started, one or plural nodes send a request for a bus use license to the parent nodes. In Fig. 13(a), the nodes C and F generate a request for a bus use license. Further, the parent node (in Fig. 13(a), the node A) sends a request for a bus use license to a parent node (relay). Finally, this request is distributed to the route to carry out the arbitration.

[0088] The route node which receives the request determines which bus should be permitted to use the bus. The arbitration work can be carried out by the route bus only. The node which wins the arbitration is given a bus use license. Fig. 13(b) shows that the node C is given a bus use license, and the use by the node F is rejected. A DP (data prefix) packet is sent to the node which loses the arbitration, which informs the node on the rejection. The bus use request by the rejected node is kept waiting the next arbitration.

[0089] Thus, thereafter, the node which wins the arbitration to attain a bus use license can start the transfer of data.

[0090] Fig. 22 is a flowchart showing the flow of the arbitration. The arbitration will be described with reference to the drawing.

[0091] The bus is required to be in th idle state in order that a node can start the data transfer. It is recognized that the preceding data-transfer has been completed, and at present, the bus is in the available state. For this purpose, each node decides that it can start the transfer from itself, based on the elapse of a predetermined time gap length (e.g., sub-action gap) which is individually set every transfer mode.

[0092] At step S401, it is determined whether a predetermined gap length is attained or not, which corresponds to data to be transfer such as Async data, Iso data, or the like. A bus use license, which is required to start the transfer, can not be requested as far as the predetermined gal length is not attained. Thus, the request is kept waiting for the time when the predetermine gap length is attained.

[0093] If the predetermined gap length is attained at the step S401, it is determined whether data to be transferred exists or not at step S402. If the data exists, a request for a bus use license is sent to the route so that the bus is secured to transfer the data at step S403. A signal for representing the request for a bus use license is transmitted while it is relayed by the respective devices in the network, and distributed to the route. If no data to be transferred exists at the step S402, the node keeps awaiting.

[0094] Then, if the route receives, at step S404, more than one bus use requests transferred at the step S403, the route checks the number of the nodes which has sent the bus use requests. If the value selected at the step S405, i.e., the number of nodes is 1 (the number of nodes which have sent the bus use requests is one), immediately, the bus use license is given to the node. If the value selected at the step S405, i.e., the number of nodes is more than 1 (the number of nodes which have sent the bus use requests is at least two), the route makes the arbitration for determining one node to be given the bus use license. The arbitration is unbiased. The arbitration is formed so that the bus use license can be equally given to the nodes without the same node being given the license, whenever the arbitration is made.

[0095] At step S407, as to the at least two nodes which make the use requests at step S406, one node which acquires the use license as a result of the arbitration by the route, and the other nodes which have lost the arbitration are selected. At step S408, the route sends a license signal to the one node which acquires the bus use license as a result of the arbitration, or the node which acquires the bus use license without the arbitration, since the value selected at the step S405, i.e., the number of nodes which make the use requests, is one at the step S405. The node which receives the license signal starts to transfer transfer-data (packet) immediately after the node receives the signal. At step S409, the route sends a DP (data prefix) packet for representing the arbitration failure to the nodes which lose the arbitration at the step S406 not to be given the license. For the nodes which receive the DP packet, the process is returned to the step S401, and await that the predetermined gap length is attained in order to make an bus use request again and transfer the data.

[0096] Thus, the flowchart of Fig. 22 showing the flow of the arbitration is described.

[0097] << Asynchronous (not synchronous) transfer >> The asynchronous transfer means non-synchronous transfer. Fig. 14 shows the time-dependent transition of the asynchronous transfer. In Fig. 1, the sub-action gap shown at the beginning represents the idle-state of the bus. when the idle time reaches a predetermined value, nodes which want to transfer data decides that the bus is available. The nodes execute the arbitration in order to acquire the bus.

[0098] After the node acquires the bus use license in the arbitration, the data transfer is executed in a packet mode. After the data is transferred, the node which receives the data sends ack (reception recognition answer code), a result of the reception, after a short gap called ack gap, for answering or response. Otherwise, the node sends a response packet. Thus, the transfer is completed. The ack comprises 4-bit information and 4-bit check-sum, and includes information for representing the success of transfer, busy-state, or pending-state, and is immediately send back to the source node.

[0099] Fig. 15 shows an example of a packet format for asynchronous transfer. The packet contains a header in addition to a data portion and data CRC. As shown in Fig. 15, destination-node ID, source-node ID, a transfer data length, various codes, and so forth are written in the header to transfer the data.

[0100] The asynchronous transfer is communication made one-on-one from a node to its destination node. A packet transferred from a source node is distributed to the respective nodes in the network. The nodes having addresses different from the address of the destination node ignore the packet. Thus, one destination node reads the packet.

[0101] The asynchronous transfer is described above.

[0102] << Isochronous Transfer >> The isochronous transfer is synchronous. The isochronous transfer, which is a greatest characteristic of the 1394 serial bus, is a transfer mode suitable for transferring data such as multi-media data, e.g., VIDEO image data and speech data which are required to be transferred in real time.

[0103] The asynchronous transfer (not synchronous) is one-on-one transfer. On the other hand, by the isochronous transfer, data is equally transferred from one source node to all of the other nodes, due to the broadcast function thereof.

[0104] Fig. 16 shows the time-dependent transition of the isochronous transfer.

[0105] The isochronous transfer is carried out via the bus at constant time-intervals. This time-interval is called an isochronous cycle. The isochronous cycle time is 125 μs. A cycle start packet has a role for indicating the start time of the cycle and adjusting the time of the respective nodes. A node called a cycle master transmits the cycle start packet. The time-interval at which cycle start packets are transmitted is 125 μs.

[0106] Plural types of packets are given IDs, and can be separately transferred in one cycle as shown at a channel A, a channel B, and a channel C in Fig. 16. Thereby, real-time transfer can be made between plural nodes at the same time. Moreover, a node to receive data captures only the data with the channel ID which the node wants to receive. The channel IDs, not representing the source addresses, are only numbers given to respective data. Accordingly, one packet, when it is transmitted, is distributed to all of the other nodes.

That is, the packet is transferred by broadcast.

[0107] The arbitration is carried out prior to the packet transmission by the isochronous transfer, which is similar to the asynchronous transfer. However, this transmission is not one-on-one communication, differently from the asynchronous transfer. Therefore, no ack (reception recognition answer code) exists in the isochronous transfer.

[0108] An iso gap (isochronous gap) shown in Fig. 16 represents an idle time period which is required to recognize that the bus is in the available state before the start of the isochronous transfer. When the predetermined idle time-period lapses, nodes which want to carry out the isochronous transfer decide that the bus is available, and can make the arbitration prior to the transfer.

[0109] Hereinafter, an example of a packet format for the isochronous transfer shown Fig. 17 will be described.

[0110] Each of the different packets, separated into the respective channels, contains a header in addition to a data portion and data CRC. As shown in Fig. 17, a transfer data length, a channel No, various codes, and error-correction header CRC are written in the header to transfer the data.

[0111] The isochronous transfer is described above.

[0112] << Bus Cycle >> In practical 1394 serial bus transfer, the isochronous transfer and the asynchronous transfer are compatible with each other. Fig. 18 shows the tie-dependent transition of the transfer in the bus, in which both of the isochronous transfer and the asynchronous are carried out.

[0113] The isochronous transfer is executed in preference to the asynchronous transfer. The reason is as follows. That is, after the cycle start packet is transmitted, the isochronous transfer can be started after a gap length (sub-action gap) which is shorter than the gap length between the idle time-periods which is required to start the asynchronous transfer.

[0114] In the general bus cycle shown in Fig. 18, when a cycle # m is started, a cycle start packet is transferred to the respective nodes from a cycle master. Thereby, each node adjusts clock-time. After the predetermined idle time-period (isochronous gap), a node which is to execute the isochronous transfer makes the arbitration, and enters the packet transfer. In Fig. 18, a channel e, a channel s, and a channel k are sequentially transferred.

[0115] The processed ranging from the arbitration to the packet transfer are repeated in correspondence to the number of channels. After the isochronous transfer is the cycle # m is completed, the asynchronous transfer can be carried out.

[0116] When the idle time-period reaches the sub-action gap at which the asynchronous transfer can be made, the nodes which want to carry out the asynchronous transfer decide that the arbitration can be executed.

[0117] It is to be noted that the asynchronous transfer can be made only in the case where the sub-action gap for starting the asynchronous transfer is obtained during the course of from the completion of the isochronous transfer to the time (cycle synch) at which the next cycle start packet is to be transmitted.

[0118] in the cycle # m of Fig. 18, the isochronous transfer in an amount of three channels is carried out, and thereafter, two-packets (packets 1 and 2) are asynchronously transferred (including ack). After the asynchronous packet 2 is carried out, the time reaches the time (cycle sync) for starting a cycle m + 1. Thus, the transfer in the cycle # m is completed then.

[0119] In the case where the time (cycle synch) at which the next cycle start packet is to be started arrives during the asynchronous or isochronous transfer, the cycle start packet for the next cycle is transmitted after the elapse of the idle time-period from the completion of the transfer, not forcedly interrupting the transfer. In particular, for example, when one cycle lasts at least 125 μs, the next cycle time-period is shortened to be less than the reference time-period, i.e., 125 μs. As seen in the above-description, the isochronous cycle can be lengthened or shortened with respect to the reference time-period, i.e., 125 μs.

[0120] However, in some cases, the isochronous transfer does not fail to be carried out every cycle in the case where the transfer must be made in real time. The asynchronous transfer is carried out in the next cycle, since the cycle time is shortened.

[0121] The above-described processes including such delay-information are controlled by the cycle master.

[0122] The IEEE 1394 serial bus is described above.

[0123] Hereinafter, a first embodiment will be described, in which devices are connected to each other via 1394 serial bus cables as shown in Fig. 2. A reference numeral 101A designates a printer device which operates as a direct printer or a network printer, 102 VTR (digital video with a camera) connected to the printer 101 via a 1394 serial bus cable, from which image data can be directly printed by means of the printer 101. Also, the VTR 102 can transfer image data or the like to other devices connected via the printer 101. Reference numeral 103 designates a personal computer (hereinafter, abbreviated as PC) connected to the printer 101 via a 1394 serial bus, and 104 does a scanner connected to the PC via a 1394 serial bus. It is to be noted that the network is a typical device group. Thus, devices may be connected on the backside of the PC 103 or the scanner 104. As a device to be connected, any device such as external memories, e.g., hard disks or the like, and CD, DVD or the like may be employed, provided that the devices can constitute a network via 1394 serial buses.

[0124] The operation of the first embodiment of the present invention using a network configuration as shown in Fig. 2 will be described with reference to Fig. 1.

[0125] Hereinafter, Fig. 1 (block diagram) will be explained.

[0126] In Fig. 1, reference numeral 1 designates the body of a printer device, 2 does the body of VTR, and 3 does a magnetic tape as a recording medium. In the present invention, other media may be used in addition to such a tape. Reference numeral 4 designates a read/write head, 5 does a regeneration-processing circuit, 6 does am image decoding circuit, 7 a D/A converter, 8 does EVF for checking an image to be played or an image to be to be printed by means of the printer. 9 does an external output terminal, 10 does an operating portion for inputting an instruction, 11 does a system controller for VTR, 12 does a frame memory, 13 does a 1394 interface (I/F) portion of the printer, 14 does a selector for plural types of data, 15 does a display circuit which enables printer information to be displayed on EVF, 16 does an image synthesizer, 17 does a 1394 interface (I/F) portion of the printer, 18 does an image processing circuit for image-forming an image to be printed in a printer which includes binarization, color correction, and so forth. Reference numeral 19 designates a memory for forming image data into a print image, 20 does a printer head, 21 does a driver for driving a printer head 20, carrying out paper-conveyance, and so forth, 22 does an operating portion for controlling the operation of the printer, 23 does a printer controller which is a control unit for the printer, 24 does a printer information generating circuit for generating printer information of the operational state of the printer in the direct print operation, and 25 does a data selector. Reference numeral SW1 designates a switch of which the ON/OFF is controlled by the printer controller 23.

[0127] In Fig. 1, for the VTR, only the regeneration system is illustrated. Moreover, the PC and the scanner shown in Fig. 2 are not illustrated in Fig. 1 for easy understanding of the description.

[0128] Hereinafter, the operation illustrated in the block diagram of Fig. 1 will be described in order.

[0129] First, image data recorded in the magnetic tape 3 is read by the read/write head 4. The read image data is processed with respect to a regeneration form in the regeneration-processing circuit 5. The read image data is encoded and recorded by a predetermined compression mode based on DCT (discrete cosine transformation) and VLC (variable length coding) which are methods of band-compressing home-digital video. Accordingly, the image data is subjected for predetermined decoding in the image decoding circuit 6, is restored to an analog signal by the D/A converter 7, and is displayed on the EVF 8 or is analog-output via the external output terminal 9 to an external device.

[0130] In the case where desired image data is transferred to another mode via the 1394 serial bus, the image data decoded in the decoding circuit 6 is temporarily stored in the frame memory 12, and is transmitted to the 1394 interface (I/F) portion 13 via the data selector 14. Then, the image data is transferred to the printer 1 in the isochronous mode. If the transferred data is to be directly printed, the printer 1 captures the image data therein. If the image data is to be transferred to another node, the image data is transferred to the target node, passing through a 1394 I/F potion 17 without being processed therein.

[0131] The input of an instruction for the VTR such as regeneration-operation or the like of the VTR is carried out by the operating portion 10. Moreover, the input of an instruction to the printer in the direct print operation can be made by the VTR operating portion 10. Based on the input of the instruction from the operating portion 10, the system controller 11 controls the respective operation portions, typically, controls the regeneration-processing circuit of the VTR. In addition, depending on the input of a predetermined instruction, the system controller 11 generates a control command for the printer, which is transferred as command data through the 1394 interface (I/F) portion 13 to the printer via the selector 14.

[0132] Referring to printer information data transmitted from the printer 1 via the 1394 serial bus, such as the operation state of the printer, warning messages, print image information, and the like, the data is transmitted via the 1394 interface (I/F) portion 13 and selector 14, is processed in the printer information display processing circuit 15 into a display-capable mode, is synthesized in the image synthesizer 16 so that the data is synthesized with an image under display on th EVF and is displayed, and is message-displayed on the EVF 8.

[0133] A switching circuit, not the image synthesizer 16, may be provided to have such a configuration that both of the display information are selectively displayed.

[0134] The data selector 14 and the data selector 25 of the printer 1 are select data to be output or input, so that respective data are sequentially distinguished and input or output in predetermined blocks.

[0135] Referring to the operation of the printer 1, data input to the 1394 I/F potion 17 are classified into the respective types of data, in the data selector 25. Data to be printed is input to the image processing circuit 18, and is image-processed so as to be suitable for printing. The data formed as print image in the readout memory 19 and controlled in the storage and readout is transferred to the printer head 20 to be printed. The driver 21 drives the printer head, the paper-feeding, or the like. The printer controller 23 control the operation of the driver 21 and the printer head 20.

[0136] The printer operating portion 22 inputs an instruction on operations such as paper-feeding, resetting, ink-checking standby/stopping of printing, and the like. The printer controller 23 controls the respective portions corresponding to the input of an instruction. Normally, the switch SW1 is in the closed state. The switch SW1, when it receives a predetermined command from the printer controller 23, opens, under predetermined conditions, the connection for all of the key-input or a part of the key-input carried out in the operating portion 22, so that all commands or a particular command can not be input for instruction. The printer controller 23 itself may be so configured that such a command can not be input for instruction, without the switch SW1 being provided.

[0137] In the case where data input to the 1394 I/F potion 17 is data representing a command for the printer 1 which is generated in the VTR 2 or the like, the data is transmitted as control command through the data selector 25 to the printer controller 23, and thus, the printer controller 23 controls the respective portions.

[0138] Messages showing the operation-state of the printer, messages showing that the printing can be finished or started, warning messages showing jamming, defective operation, ink-deficiency, or the like, print image information, and so forth, generated in the printer information generating circuit 24, are input to the data selector 25, and then, can be output via the 1394 I/F potion 17 to the outside. In the VTR 2, the output printer information as base can be processed for display in the printer information display processing circuit 15.

[0139] A user observes the messages or the print image information displayed on the EVF 8, based on the printer information, and then, inputs a command for the printer 1 to take appropriate countermeasures in the operating portion 10. Thus, the control command data is transmitted via the 1394 serial bus, so that the operation of the respective portions of the printer 1 and an print image in the image processing circuit 18 can be controlled by the printer controller 23.

[0140] As described above, image data and different command data are transferred, if necessary, via the 1394 serial bus connecting the VTR 2 and the printer 1 between them.

[0141] Referring to the transfer mode in which respective data are transferred from the VTR 2, in most cases, image data (and audio data) are transferred as Iso data via the 1394 serial bus according to the isochronous transfer mode, while command data are transferred as Async data according to the asynchronous transfer mode in conformity to the specifications of the above--described 1394 serial bus. However, in some cases, it is preferable to transfer some types of data in the asynchronous transfer mode rather than in the isochronous transfer mode. In such cases, the asynchronous transfer mode is employed.

[0142] Printer information to be transferred from the printer 1 are transferred as Async data in the asynchronous transfer mode. However, as for print image information or the like, of which the amount is large, the information may be transferred as Iso data in the isochronous transfer mode.

[0143] The block diagram of Fig. 1 is described above. Needless to say, in the case where a network as shown in Fig. 2 is configured by use of the 1394 serial bus, respective data can be bidirectionally transferred between the VTR 2, the printer 1 and the PC 103, the scanner 104, based on the specifications of the 1394 serial bus.

[0144] When the direct printing of image data is carried out by the VTR 2 and the printer 1 in the configuration shown in the block diagram of Fig. 1, generally, it is preferable that the operation of the printer 1 can be controlled by the manipulation of the VTR 2 only. This is possible in the configuration of Fig. 1. Thus, it is made possible to control the respective portions of the printer 1 by the manipulation of the VTR 2 only, when the direct print operation is carried out. According to the present invention, at this time, the input of any instruction or the input of a particular command with respect to the operation of the printer 1, made in the operating portion 22, is disabled (not accepted). Thereby, the system can be set so as to eliminate various error operations at direct printing. This can be realized by the fact that the printer 1 receives data for representing the start of direct print, which is transferred from the VTR 2 via the 1394 serial bus before image data to be directly printed is transferred, and the switch SW1 is opened due to the control by the printer controller 23.

[0145] The system is set so that the mutual recognition of the start of the direct print operation (mode) between the VTR 2 and the printer 1 starts when a direct print start signal is transmitted and received, while the direct print mode ends when the a direct print end data is transferred from the VTR 2 via the 1394 serial bus and received by the printer 1, or when the connection of the 1394 serial bus connecting the VTR 2 and the printer 1 between them is cancelled, and moreover, the printer controller 23 re-starts the connection of the switch SW1 when the controller determines that the direct print mode is finished. The printer device can automatically determines whether the connection of the 1394 serial bus is disabled or not, based on the bus-rest of the 1394 serial bus and the formation of a new bus configuration.

[0146] As a switch device of the operating portion 10 of the VTR 2 for output a command to instruct the stat/end of the direct print, a switch shown in Fig. 5 is fixed. In Fig. 5, "OFF" represents the OFF state of the electric source, "Photograph" represent a position at which an image or speech is recorded, and "Regeneration" normally represents a position at which the regeneration is carried out. Moreover, "Direct Print" represents a position at which the direct print mode is carried out. When the switch is turned to the position, the VTR 2 transmits a direct print stat command to the printer. When the switch is shifted from the above-mentioned position, the VTR 2 transmits a direct print end command. A "Push" key located in the center may play a role as a trigger for photographing, as a command input switch for optional selection, as an image data transmission start switch for direct print or the like, and so forth.

[0147] Moreover, the system may be set as follows. That is, it is not necessary to transmit a command representing the start/end of direct print separately from image data. When image data is transferred from the VTR 2 to the printer 1, the direct print may be started, based on the decision on the header information of a packet containing image data.

[0148] Hereinafter, the sequence in which the input of an instruction by the operating portion 22 of the printer in the direct print operation mode is disabled is described, including the flow of the operations of the VTR 2 and the printer 1 when the direct print is carried out, in reference to the flowchart of Fig. 6.

[0149] First, at step S1, the printer operating portion 22 is in the instruction input-capable state when the system is in the normal mode. thus, th switch SW1 is in the connected (ON) state. Then, at step S2, when a user shifts the system to the direct print mode, the direct print start command is transmitted from the VTR 2 in the above-described manner. The transmitted start command is transferred via the 1394 serial bus in an asynchronous packet, and the printer 1 receives the command at step S3. Thereby, the printer controller 23 causes the switch SW1 to open (OFF) for the start of the direct print mode at step S4. In this manner, one of the switch SW1, the operating portion 22, and the printer controller 23 is formed so that the input of any instruction or the input of a particular instruction by the operating portion 22 is interrupted, disabled, and ignored.

Thus, the input of a particular instruction can not be accepted, disabled, or ignored.

[0150] It is to be noted that if no start command is received at the step S3, the connection of the switch SW1 is maintained until the start command is received.

[0151] In the VTR 2, an image to printed is selected by a user in parallel to the process for shifting to the direct print mode of the printer 1. At step S5, an optional image is selected for transfer. At step S6, the specified image is data is transferred via the 1394 serial bus in an isochronous (or asynchronous) packet. Then, the process moves to the step S9. If no selection of an optional image is made at step S5, the transfer of image data is not carried out, and the process moves to the step S9.

[0152] If the printer 1 receives image data packet-transferred via the 1394 serial bus at step S7, the image data is printed in the predetermined sequence at step S8. Thereafter, the process is returned to the step s7, where the reception of the next image data is accepted.

[0153] In the VTR, at step S9, a user selectively finishes the direct print mode, or specifies another image for transfer, not finishing the direct print mode. If the user wants to specify another image, not finishing the direct print mode at the step S9, the process is returned to the step S5, where the user can specify an optional image again.

In this case, the image-specification at the step S5 and the transfer of the image data at the step S6 can be repeated via the step S9. In the case where printing is made on plural sheets, the transfer is carried out under the control of the output in response to the operation speed of the printer 1.

[0154] If the user selects the completion of the direct print mode at the step S9, a direct print end command is transmitted at step S10. The end command data is transferred via the 1394 serial bus, and then, the direct print mode of the VTR is completed.

[0155] If the printer 1 receives no image data to be printed, from the VTR 2 at the step S7, the process moves to step S11, where the direct print end command is accepted. If no direct print end command from the VTR 2 is accepted at the step S11, the process is returned to the step S7, where the direct print mode is maintained, and image data to be printed, transmitted from the VTR 2 is accepted.

[0156] If the direct print end command data transferred in a packet from the VTR 2 at the step S10 is received at the step S11, the printer controller 23 controls the switch SW1 to be connected (ON) at step S12, so that the direct print mode in the printer 1 is completed, and is restored to the ordinary operation mode. Thereby, the direct print mode of the VTR 2 and the printer 1 is finished.

[0157] The flowchart of Fig. 6 is described above.

[0158] If the cable of the 1394 serial bus connecting the VTR 2 and the printer 1 between them is disconnected in the direct print mode for some reason, the printer can automatically determine that it is not connected to the VTR 2 based on the generation of bus-reset and the formation of a new network.

[0159] The first embodiment has been described.

[0160] <Second Embodiment> The second embodiment of the present invention in which the VTR 102 shown in Fig. 2 is replaced by a digital camera is described.

[0161] Fig. 4 is a block diagram of the present invention in which the digital camera and the printer are connected to each other via the 1394 serial bus cable, which will be described below.

[0162] In Fig. 4, reference numeral 1 designates the body of a printer device, 61 does the body of the digital camera, 62 does an image pick-up portion, 63 does an A/D converter, 64 does an image processing portion, 65 does an image-encoding/decoding circuit, 66 does a memory recording regeneration portion for recording/regenerating an image, 68 does EVF which is a display portion, 69 does a digital camera operating portion, 70 does a system controller for the digital camera, 71 does a data selector, 72 does the 13941 I/F portion for the digital camera, 73 does a display-processing circuit for printer information, and 74 does an image synthesizer.

[0163] The printer 1 is the same as that described in the first embodiment. However, in the configuration of the printer 1, a decoding circuit 25 is provided between the data selector 25 and the image processing circuit 18.

[0164] In the image-encoding/decoding circuit 65 of the digital camera 61, an image is encoded by a JPEG format known as a technique for encoding a still picture.

[0165] Hereinafter, the operations illustrated in the block diagram of Fig. 4 will be described in order.

[0166] First, image data picked up in the image pick-up portion 62 at recording by the digital camera 61 is digitalized by the A/D converter 63, and is processed for the image data in the A/D converter 63 so as to be suitable for displaying. One output from the image processing portion image processing portion 64 is returned to the analog signal as an image under image picking-up by the image processing portion 64, and is displayed on the EVF 68. Th other output is encoded in the JPEG format in the encoding circuit 65, and is stored in the memory of the memory recording portion 66.

[0167] For regeneration, a desired image is read in the memory regeneration portion 66. At this time, a desired image is selected based on the information input from the operating portion 69, and is read under the control by th system controller 70. For the image data regenerated from the memory, the JPEG compression is decoded in the decoding circuit 65, and can be processed in the image processing portion 64 and a D/A converter 67 to be displayed on the EVF 68.

[0168] In the case where desired image data are regenerated from the memory, and are transferred for direct print or to PC for other devices which is connected via the 1394 serial bus, the image data is transferred through the data selector 71 from the 13941 I/F portion 72 via the 1394 serial bus. At this tie, the image data regenerated in the memory regeneration portion 6 and encoded in the JPEG format is output as it is. For direct print, the image data is decoded in the printer.

[0169] The operation of the printer 1 is similar to that in the first embodiment, and is not described here. The decoding circuit 26 will be described. For image data transferred from the digital camera 61, the JPEG compression data is decoded by use of software in the decoding circuit 26. Referring to the decoding circuit 26, the image data is processed in a circuit within the printer device or is processed and decoded by use of software in CPU, using a JPEG decoding program file held in ROM or decoding date transmitted together with the compressed image data from the digital camera 2.

[0170] The image data compressed in the JPEG format is transferred from the digital camera to the printer, and is decoded within the printer, as described above. Thus, the transfer rate is higher as compared with the transfer after the image date is converted to the non-compressed data. Moreover, the JPEG decoding can be made by use of software. Even if a decoder is provided for the printer, conveniently, the cost is not high. For the decoding circuit 26, a JPEG decoding circuit (board) may be provided as decoding using hardware.

[0171] The input of instructions to the respective portions of the digital camera is made from the operating portion 69. Moreover, the input of an instruction to the printer in the direct print operation can be made by the camera operating portion 69. Based on the input of the instruction from the operating portion 69, the system controller 70 controls the respective operation portions, typically, controls the regeneration processing circuit of the digital camera. In addition, depending on the input of a predetermined instruction, the system controller 70 generates a control command for the printer, which is transferred as command data through the data selector 71 and the 13941 I/F portion 72 to the printer.

[0172] Referring to printer information data transmitted from the printer 1 via the 1394 serial bus, such as the operation state of the printer, warning messages, print image information, and the like, the data is transmitted via the 1394 (I/F) portion 72 and the selector 71, is processed in the printer information display processing circuit 73 into a display-capable form, is synthesized in the image synthesizer 74 so that the data is synthesized with an image under display on the EVF, and is message-displayed on the EVF 68. Alternatively, a switching circuit, not the image synthesizer 74, may be provided to have such a configuration that both of the display information are selectively displayed.

[0173] The data selector 71 selects data, so that the respective data are sequentially distinguished foe each type, and are input or output to or from predetermined blocks.

[0174] Referring to the transfer mode in which respective data are transferred from the digital camera 62, in most cases, image data are transferred as Iso data via the 1394 serial bus according to the isochronous transfer mode, while command data are transferred as Async data according to the asynchronous transfer mode in conformity to the specifications of the above-described 1394 serial bus. However, in some cases, it is preferable to transfer data in the asynchronous transfer mode rather than in the isochronous transfer mode. In such cases, the asynchronous transfer mode is employed.

[0175] The block diagram of Fig. 1 is described above. Needless to say, in the case where another device together with the printer 1 forms a network via the 1394 serial bus, respective data can be bidirectionally transferred between the digital camera 61, the printer 1 and the PC 103, the scanner 104, based on the specifications of the 1394 serial bus.

[0176] When the direct printing of image data is carried out by the VTR 2 and the printer 1 in the configuration shown in the block diagram of Fig. 4, preferably, the system is set so that the operation of the printer 1 can be controlled by the manipulation of the VTR 2 only. In this case, according to this embodiment, the input of any instruction or the input of a particular command with respect to the operation of the printer 1, transmitted from the operating portion 22, is disabled (not accepted). Thereby, the system can be set so as to eliminate various error operations during the direct printing. This can be realized as follows. That is, a switch as shown in Fig. 5 is provided as a part of the instruction input keys in the operating portion 62 of the digital camera 61. The printer 1 receives a packet for representing the start of direct print, which is transferred from the digital camera 61 via the 1394 serial bus before image data to be directly printed is transferred. Then, the switch SW1 is opened under the control by the printer controller 23.

[0177] Moreover, a part of the operation information can be disabled by the processing in the operating portion 22 and the printer controller 23, using software.

[0178] The system is set as follows. That is, the mutual recognition of the start of the direct print operation (mode) between the VTR 2 and the printer 1 starts when a direct print start signal is transmitted and received, and the direct print mode ends when a packet of direct print end data is transferred from the digital camera 61 via the 1394 serial bus (Async) and received by the printer 1, or when the connection of the 1394 serial bus connecting the digital camera 61 and the printer 1 between them is cancelled. The printer controller 23 re-starts the connection of the switch SW1 when the controller determines that the direct print mode is finished. The printer device can automatically decide that the connection of the 1394 serial bus is disabled, based on th generation of bus-reset of the 1394 serial bus and the formation of a new bus configuration.

[0179] The systematical operation of the printer 1, and also the systematical operation of the digital camera 61 and the printer 1 during the direct print according to the present invention are the same as those described in the first embodiment, and can be understood in the flowchart of

Fig. 6. Thus, the operations are not described here.

[0180] The second embodiment is described above.

[0181] <Other Embodiments> The printer is controlled by use of a visual interface (so-called GUI) which is generally used as a monitor of PC. However, all of the functions can not be monitored by DUI or the like, when the direct print is carried out. Accordingly, the system may be set so that as to only the functions that can be supported on the camera side, the operation is enabled, and as to only the functions that can not be monitored, the operation disabled.

[0182] As described above, according to this embodiment, the print-process which a user wants to carry out in preference can be quickly performed according to the direct print process.

[0183] If a command from the camera or VTR is accepted in the direct print operation, so that the print operation is carried out, the input of a command by the instructing portion provided for the printer is disabled. Thus, error operations during the direct print are eliminated, or error operations can be reduced.

[0184] As for the direct print using the 1394 serial bus, data communication can be made not by way of PC to output an image print. Therefore, the process can be performed at high speed, not affected by the operation-conditions of the

PC. In addition, the charge to the PC, which will be caused by the print data processing, can be eliminated.

[0185] The error operations of the printer can be reduced by disabling only the functions of which the operation-state can not be confirmed. Thus, advantageously, an superior interface can be also provided when the direct print is carried out.

[0186] As the interface used in this embodiment, an interface in compliance with the 13094 standards is described. Interfaces using IR or radio waves are available.

[0187] As the printer, devices using ink-jet or a electrophotographic system are available.

[0188]

[Advantages] A According to the present invention, arbitration can be made while the operation between two nodes is not instructed by each other.

[0189] Also, according to the present invention, when the direct print is carried out, a command from the camera or the printer is accepted, so that the input of a command from the instructing portion disposed in the printer is disabled. Thus, no error operation occurs, and the system can be easily used.

[Brief Description of the Drawings]

[Fig. 1] Fig. 1 is a block diagram of a printer device and

VTR to which the present invention is applied.

[Fig. 2] Fig. 2 illustrates an example of a network used to carry out the present invention.

[Fig. 3] Fig. 3 is a block diagram showing a conventional configuration in which a digital camera, PC, and a printer are connected, using the PC as a center.

[Fig. 4] Fig. 4 is a block diagram of a printer device and a digital camera to which the present invention is applied.

[Fig. 5] Fig. 5 illustrates an example of a switch used in the present invention.

[Fig. 6] Fig. 6 is a flowchart of the operations of VTR and th printer to which the present invention is applied.

[Fig. 7] Fig. 7 shows an example of a network configuration using a 1394 serial bus for connection.

[Fig. 8] Fig. 8 shows the components of the 1394 serial bus.

[Fig. 9] Fig. 9 is an address map of the 1394 serial bus.

[Fig. 10] Fig. 10 is across-sectional view of a 1394 serial bus cable.

[Fig. 11] Fig. 11 illustrates a DS-Link encoding system.

[Fig. 12] Fig. 12 illustrates the topology setting for determining the IDs of nodes in the 1394 serial bus.

[Fig. 13] Fig. 13 illustrates the arbitration in the 1394 serial bus.

[Fig. 14] Fig. 14 shows a basic constitution of the temporal state-transition of asynchronous transfer.

[Fig. 15] Fig. 15 shows an example of the format of an asynchronous-transferred packet.

[Fig. 16] Fig. 16 shows a basic constitution of the temporal state-transition of the asynchronous transfer.

[Fig. 17] Fig. 17 shows an example of the format of an isochronous-transferred packet.

[Fig. 18] Fig. 18 shows an example of a bus cycle showing the state of a packet transferred in a practical 1394 serial bus.

[Fig. 19] Fig. 19 is a flowchart of the flow ranging from the bus-reset to the determination of the node IDs.

[Fig. 20] Fig. 20 is a flowchart of the flow of the determination of a parentage at the bus-reset.

[Fig. 21] Fig. 21 is a flowchart of the flow ranging from the determination of the parentage at the bus-reset to the determination of the IDs.

[Fig. 22] Fig. 22 is a flowchart showing the arbitration.

[Reference Numerals]

1; printer device

2; VTR

10; operating portion (VTR)

11; system controller (VTR)

22; operating portion

23; printer controller

24; printer information generating portion

62; operating portion (digital camera)

70; system controller (digital camera)

SW1; switch

Fig. 1

5 REGENERATION-PROCESSING CIRCUIT IMAGE 6 DECODING 10 OPERATING PORTION 11 SYSTEM CONTROLLER 12 FRAME MEMORY 14 DATA SELECTOR 15 PRINTER INFORMATION DISPLAY PROCESSING CIRCUIT 13 1394 INTERFACE (I/F) PORTION 9 EXTERNAL OUTPUT 1394 CABLE TO PC 17 1394 I/F POTION 25 DATA SELECTOR 22 OPERATING PORTION 23 PRINTER CONTROLLER 19 MEMORY 18 IMAGE PROCESSING 21 DRIVER 20 PRINTER HEAD 24 PRINTER INFORMATION GENERATING CIRCUIT

FIG. 2

1394 SERIAL BUS 104 SCANNER 1394 SERIAL BUS 101 PRINTER 1394 SERIAL BUS 102 VTR(DIGITAL VIDEO)

FIG. 3

42 DECODING 43 DISPLAY 44 HARD DISK 45 MEMORY 47 PCI BUS 41 OPERATING PORTION 40 DIGITAL I/O 34 MEMORY 39 DIGITAL I/O 35 DECODING 36 IMAGE PROCESSING PORTION DIGITAL CAMERA 52 PRINTER CONTROLLER 53 DRIVER 51 PRINTER HEAD 50 MEMORY

FIG. 4

DIGITAL CAMERA 62 IMAGE PICK-UP PORTION IMAGE PROCESSING PORTION 64 IMAGE PROCESSING PORTION 65 ENCODING/DECODING 66 MEMORY RECORDING REGENERATION PORTION 73 PRINTER

INFORMATION DISPLAY-PROCESSING 71 DATA SELECTOR 72 13941 I/F PORTION 70 SYSTEM CONTROLLER 69 OPERATING PORTION 1349 CABLE TO PC 17 1394 I/F PORTION PRINTER 25 DATA SELECTOR 22 OPERATING PORTION 26 DECODING 23 PRINTER CONTROLLER 19 MEMORY 18 IMAGE PROCESSING 21 DRIVER 20 PRINTER HEAD 24 PRINTER INFORMATION GENERATING

FIG. 5

IMAGE PICK-UP REGENERATION DIRECT PRINT

FIG. 6

PRINTER-START S1 NORMAL MODE SW1 = ON S3 START-COMMAND RECEIVED ? S4 DIRECT PRINT MODE SW1 = OFF S7 IMAGE-DATA RECEIVED ? S8 PRINT S2 DIRECT PRINT START COMMAND S5 IMAGE IDENTIFIED ? S6 TRANSFER IDENTIFIED IMAGE S9 FINISHED ? S10 DIRECT PRINT END COMMAND S11 END COMMAND RECEIVED ? S12 NORMAL MODE SW2 = ON PRINTER-END

FIG. 7

DEVICE A DEVICE B DEVICE C DEVICE D DEVICE E DEVICE F DEVICE G DEVICE H 1394 SERIAL BUS

FIG. 8

APPLICATION LAYER SERIAL BUS MANAGEMENT TRANSACTION LAYER LINK LAYER PHYSICAL LAYER 1394 CONNECTOR PORT CABLE

FIG. 9

BUS #0 BUS #1 BUS #1022 BUS #1023(LOCAL BUS) NODE #0

NODE #1 NODE #62 NODE #63 (BROADCAST) BUN NO. NODE NO.

10 BITS 10 BITS INHERENT DATA REGION (28 BITS) 48 BITS

IMAGE PROCESSING PORTION 64 BITS

FIG. 10

ELECTRIC POWER LINE (8 TO 40 V, DC MAX. CURRENT 1.5 A)

TWISTED PAIR SIGNAL LINE 2 SETS (CABLE CROSS-SECTION)

SIGNAL LINE SHIELD

FIG. 11

(SIGNAL OF EXCLUSIVE OR OF DATA AND STROBE)

FIG. 12

<ROUTE> NODE B BRANCH LEAF C NODE A BRANCH NODE C

BRANCH NODE D LEAF NODE F LEAF NODE E

BRANCH: NODE HAVING AT LEAST TWO NODE-CONNECTIONS

LEAF: NODE HAVING ONLY ONE CONNECTED PORT

PORT

c: port corresponding to child node

p: port corresponding to parent node

Fig. 13

1 ROUTE 2 REQUEST 3 REQUEST 4 REQUEST (a) REQUEST FOR BUS USE LICENSE 5 ROUTE 6 REJECTION (DP) 7 REJECTION (DP) 8 PERMISSION 9 REJECTION (DP) (b) PERMISSION OF BUS USE

Fig. 14

1 ARBITRATION 2 PACKET TRANSFER

FIG. 15

1 TOTAL 24 + 4 X N BYTES 2 DESTINATION NODE ID 3 SOURCE NODE ID 4 DATA LENGTH 5 HEADER CRC 6 DATA FIELD (4 X N BYTES) 7 DATA CRC

FIG. 16

CHANNEL A CHANNEL B CHANNEL C ONE CYCLE 125 μS CYCLE START PACKET ARBITRATION PACKET TRANSFER

FIG. 17

1 TOTAL 12 + 4 X N BYTES 2 DATA LENGTH 3 CHANNEL NO. 4 HEADER CRC 5 DATA FIELD (4 X N BYTES) 6 DATA CRC

Fig. 18

1 CYCLE TIME = 125 μS (NOMINAL) 2 (SHORT GAP) 3 (LONG GAP) 4 (SHORT GAP) 5 (SHORT GAP) 6 CYCLE # m 7 (CYCLE START PACKET) 8 ISOCHRONOUS PACKET (ch: CHANNEL) 9 ASYNCHRONOUS PACKET 10 acl (ACKNOWLEDGEMENT)

FIG. 19

S101 BUS RESET ? S102 DECLARE PARENTAGE S103 PARENTAGES OF ALL NODES DETERMINED ? S104 DETERMINE ROUTE S105 SET NODE ID S106 ID SETTING FINISHED ? S107 TRANSFER DATA

FIG. 20

S201 BUS RESET ? S202 FL = LEAF S203 RECOGNIZE PORT S204 NUMBER OF UNDETERMINED PORTS = ? S208 FL = ROUTE S209 RECOGNIZE ROUTE S205 DECLARE CHILD S206 FL = BRANCH S207 ACCEPT PARENT

FIG. 21

ROUTE BRANCH LEAF S302 SET NUMBER OF LEAVES (= N) S303 REQUEST ID S304 ARBITRATION S305 POST RESULT S306 ID ACQUIRED ? S307 BROADCAST ID INFORMATION S308 COUNT N = N - 1 S310 SET NUMBER OF BRANCHES (= M) S311 REQUEST ID S312 ARBITRATION S313 POST RESULT S314 ID ACQUIRED ? S315 BROADCAST ID INFORMATION S316 COUNT M = M -1 S318 ACQUIRE ROUTE ID S319 BROADCAST ROUTE ID

FIG. 22

S401 PREDETERMINED GAP LENGTH ? S402 TRANSFER DATA EXIST ? S403 REQUEST BUS USE LICENSE S404 RECEIVE ROUTE S405 NUMBER OF NODES REQUESTING USE ? S406 ARBITRATE S407

PERMITTED ? S408 TRANSMIT PERMISSION-SIGNAL S409 TRANSMIT DP